PCS 2428 / PCS 2059
Inteligência Artificial

Prof. Dr. Jaime Simão Sichman
Prof. Dra. Anna Helena Reali Costa

Tipos de Encadeamento
Prolog

Cláusulas de Horn

Inferências com Cláusulas de Horn

- Existem algoritmos especializados para realizar inferências em bases de conhecimento compostas por cláusulas de Horn
- São denominados de encadeamento para frente (forward chaining) e encadeamento para trás (backward chaining)
- Decidir se uma cláusula é consequência lógica de uma base utilizando cláusulas de Horn pode ter uma implementação <u>linear</u> em função da BC

Encadeamento para a Frente

1. p → q
2. l ∧ m → p
3. b ∧ l → m
4. a ∧ p → l
5. a ∧ b → l
6. a
7. b

---

8. a ∧ b — AND-I 6,7
9. l — MP 8,5
10. b ∧ l — AND-I 7,9
11. m — MP 10,3
12. l ∧ m — AND-I 9,11
13. p — MP 12,2
14. q — MP 13,1

Desejo provar q

- Encadeamento para a frente é um método correto e completo
- Chega num determinado **ponto fixo** onde novas inferências não são possíveis
- Exemplo de conceito de raciocínio orientado pelo dados (data driven)

Encadeamento para Trás

1. p → q
2. l ∧ m → p
3. b ∧ l → m
4. a ∧ p → l
5. a ∧ b → l
6. a
7. b

---

1. Para provar q, vou provar p
2. Para provar p, vou provar l e m
3. Para provar l, vou provar a e b
4. Para provar m, vou provar b e l
5. Tenho em BC o fato a
6. Tenho em BC o fato b
7. Provo l
8. Provo m
9. Provo p
10. Provo q

Desejo provar q

- Encadeamento para trás é um método correto e completo
- Exemplo de conceito de raciocínio orientado pelo objetivos (goal driven)
- Mais eficiente que encadeamento para a frente

Linguagens de Programação Lógica

- Prolog é o exemplo mais conhecido
- Kowalski: “Algorithm = Logic + Control”
- Programa = sequência de cláusulas de Horn
- Negação por falha : `not P` é considerado provado caso o programa falhe em provar `P`
- Predicados built-in para aritmética, entrada e saída

Linguagens de Programação Lógica

- As inferências de Prolog são realizadas com encadeamento regressivo, utilizando busca em profundidade
  - Inferência incompleta, cabe aos programadores se preocupar em não utilizar definições recursivas infinitas
- A ordem de busca é da esquerda para a direita para os conjuntos de uma premissa, e do início para o final para as cláusulas da BC
- A rotina de unificação não realiza a verificação de ocorrências internas
  - occur-check: {x/y, y/F(x)})
  - Inferência não correta, mas erros ocorrem muito raramente na prática

Negação por Falha

- Se o fato a ser provado não aparece explicitamente na base e nem pode ser deduzido por nenhuma outra regra...
- ... duas coisas podem ocorrer, dependendo da implementação do sistema
  - o fato é considerado FALSO
    - ex. Prolog
  - o sistema consulta o usuário via sua interface
    - ex. ExpertSinta